

Sample Scripts from UGIC 2011 ArcPy Workshop

Contributed by Kasey Hansen
29, Jun. 2011
Last Updated 05, Jul. 2011

Several GIS users in the community have requested that I post the scripts that were used during the ArcPy Workshop at UGIC 2011 in April. Below are two of the more useful scripts. The scripts are heavily commented to help anyone new to Python better understand what each line of code is doing.

Note that both scripts below reference datasets and pathnames used in the workshop (such as "Western_States.mxd", "Mountain_States.shp", etc.). In order to make these scripts work with your own data, you will need to adjust the dataset names and pathnames accordingly.

Please let me know if you have any questions or comments. Thanks!

The first script (Exercise 4 from the class) starts with a polygon shapefile and a list of geodatabases. The script first generates a new set of geodatabases that correspond with the originals. It then cycles through and clips each feature class in each of the original geodatabases and stores the resulting feature classes in the new geodatabases. This could be helpful if you were to receive several datasets from a federal, state, or county source and wanted to clip them all down to match your local boundaries.

```
#ArcPy Workshop
#Exercise 4
```

```
#This script will take several geodatabases populated with feature classes
#whose data covers the entire United States, and will create new geodatabases
#with matching feature classes clipped to the mountain states region
```

```
#import the arcpy, os and string modules
import arcpy,os,string
```

```
#Allow overwriting of existing data
arcpy.env.overwriteOutput = 1
```

```
#Set the clip layer to the mountain states shapefile
clip_layer = r"C:\ArcPy_Training\Data\mountain_states.shp"
```

```
#Create a list of the geodatabases in the main folder
```

```
#Variable "mainpath" contains the root data pathname
mainpath = r"C:\ArcPy_Training\Data"
```

```
#Variable "clippath" contains the pathname for where the mountain states GDB's will go
clippath = mainpath + "\\Mountain_States"
```

```
#Set the workspace to "mainpath"
arcpy.env.workspace = mainpath
```

```
#List all workspaces in the current workspace that end in .GDB (filter out folders aka Shapefile workspaces)
fgdb_list = arcpy.ListWorkspaces("*.gdb")
```

```
#Cycle through the old Geodatabases...
for fgdb in fgdb_list:
```

```
    #Create the New Geodatabase
```

```
    #Get the name of the current geodatabase
    basename = os.path.basename(fgdb)
```

```
    #Create a string "newfgdb" to represent the path of the new geodatabase
    newfgdb = string.replace(basename,".gdb","_Mountain.gdb")
```

```
    #Print the name of the geodatabase being created
```

```

print "Creating "+newfgdb+"....."

#Create the new geodatabase
arcpy.CreateFileGDB_management(clippath,newfgdb)

#Cycle through the feature classes and clip each one using the Mountain States Layer
#Set the workspace to the current original geodatabase
arcpy.env.workspace = fgdb

#List all feature classes in the geodatabase
fc_list = arcpy.ListFeatureClasses()

#Cycle through the list of feature classes
for fc in fc_list:

    #Print a note that says we are clipping the current feature class
    print "    Clipping " + fc + "....."

    #Create a string for the location of the new feature class
    new_fc = clippath+"\\ "+newfgdb+"\\ "+fc

    #Clip the feature class
    arcpy.Clip_analysis(fc,clip_layer,new_fc)

#Print DONE!!!!
print "DONE!!!!!"

```

The second script (Exercise 5 from the class) takes a version 10 mxd with data driven pages enabled and exports a pdf mapbook for every thematic layer in the map (in our class these layers included schools, churches, parks, etc.). The script assumes that the mxd has been previously saved with all thematic layers turned off, and any background layers turned on. It then cycles through the map and turns each thematic layer on, changes to the title on the layout to match, generates the pdf mapbook, turns the layer off, and moves on to the next layer.

```

#ArcPy Workshop
#Exercise 5

```

```

#This script will take an mxd and export a pdf mapbook for
#each thematic layer in the map. The subtitle of the map
#will be dynamically changed with each mapbook.

```

```

#Import the arcpy package
import arcpy

#Assign the Western States Mapping Document to variable mxd
mxd = arcpy.mapping.MapDocument(r"C:\ArcPy_Training\Map_Documents\Western_States.mxd")

#Create a data driven pages object using "mxd" and assign it to variable "ddp"
ddp = mxd.dataDrivenPages

#Create a list of all text elements in the map layout
textlist = arcpy.mapping.ListLayoutElements(mxd,"TEXT_ELEMENT")

#Cycle through the text elements
for txt in textlist:

    #When you find one whose text says "MAP SUBTITLE"...
    if txt.text == r"MAP SUBTITLE":

```

```
#....Assign it to variable "subtitle"
subtitle = txt

#Create a list of the layers in the map document
layerlist = arcpy.mapping.ListLayers(mxd)

#Cycle through each layer
for lyr in layerlist:

    #If the layer is NOT visible...
    if lyr.visible == False:

        #...Turn the layer on
        lyr.visible = True

        #Change the Subtitle text element to match the layer's name
        subtitle.text = lyr.name

        #Create a string showing the path of the pdf that will be written out
        pdfpath = "C:\\arcpy_training\\pdfs\\" + lyr.name + ".pdf"

        #Print that we are exporting a mapbook
        print "Exporting " + pdfpath + "....."

        #Export the mapbook to pdf
        mxd.dataDrivenPages.exportToPDF(pdfpath,"ALL")

        #Turn the layer off again
        lyr.visible = False

#Delete variables to clean up memory usage
del ddp, mxd

print "Done!"
```